CHAPTER

4

PLANNING THE ACTIVE DIRECTORY AND SECURITY

After reading this chapter and completing the exercises you will be able to:

- Explain the contents of the Active Directory
- Plan how to set up Active Directory elements such as organizational units, domains, trees, forests, and sites
- Plan which Windows 2000 security features to use in an organization, including interactive logon, object security, and services security
- Plan how to use groups, group policies, and security templates
- Plan IP security measures

Early networks existed in a realm of trust, similar to small communities in which people never locked their doors. Today, life is more complex, and security is as much an issue for networks as it is for large and small communities. One profound effect that you can have on your organization through server and network planning is to provide a blueprint for security that enables your organization to accomplish work in a secure but flexible context. Some organizations impose such tight security that it not only exceeds what is necessary, but limits what people can accomplish through their network. Security that is poorly planned or excessive can reduce the ability of your network and servers to help people reach their productivity goals. Other organizations ignore security until it is too late, and critical information is stolen or compromised.

In this chapter, you learn the capabilities available in Windows 2000 Server that enable you to build a network to exactly fit the security needs of your organization. Windows 2000 Server offers the Active Directory and an extensive range of security tools to help you manage network resources. The Active Directory provides a way to track all of your network resources, including servers, printers, and users. You learn how to structure the Active Directory for different types of situations from small offices to large multisite organizations. You also learn about the security tools in Windows 2000, which include logon security, rights and permissions, group policies, security templates, and IP security.

Defining the Active Directory

In Chapter 1 you learned that the Active Directory houses information about all network resources such as servers, printers, user accounts, groups of user accounts, security policies, and other information. You might think of it as the central nervous system of your Windows 2000 Server network because it has such a far-reaching impact on a network, from the smallest print request to the largest management activity.

In previous versions of Windows NT Server, some of the information now contained in the Windows 2000 Active Directory, such as information about user accounts, groups, and privileges, is stored in the Security Accounts Manager database (SAM). The SAM is kept on a main server, called the primary domain controller (PDC), and is regularly backed up on other servers called backup domain controllers (BDCs), as Figure 4-1 shows. Every Windows NT Server network can have only one PDC, but many BDCs. If the PDC fails, you manually promote a BDC to become the new PDC.

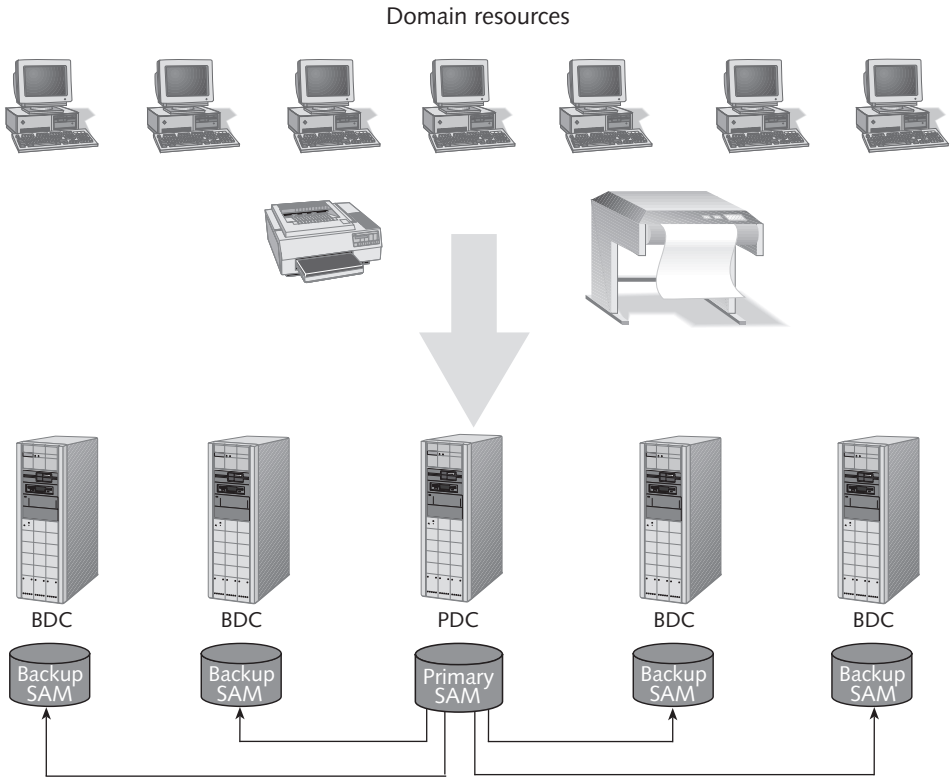

Figure 4-1 Windows NT SAM architecture

Windows 2000 Server retires the SAM and implements a broad range of directory services through the Active Directory. If you are a Windows NT Server administrator, one of the first changes you will notice is that there are no PDCs and BDCs; instead, all Windows 2000 Servers

participate in tracking network resource information via the Active Directory, and are simply called **domain controllers (DCs)**. In the Active Directory, a **domain** is a fundamental component or container that holds information about all network resources that are grouped within it—servers, printers and other physical resources, users, and user groups. Every resource is called an **object** and is associated with a domain (see Figure 4-2). When you set up a new user account or a network printer, for instance, it becomes an object within a domain.

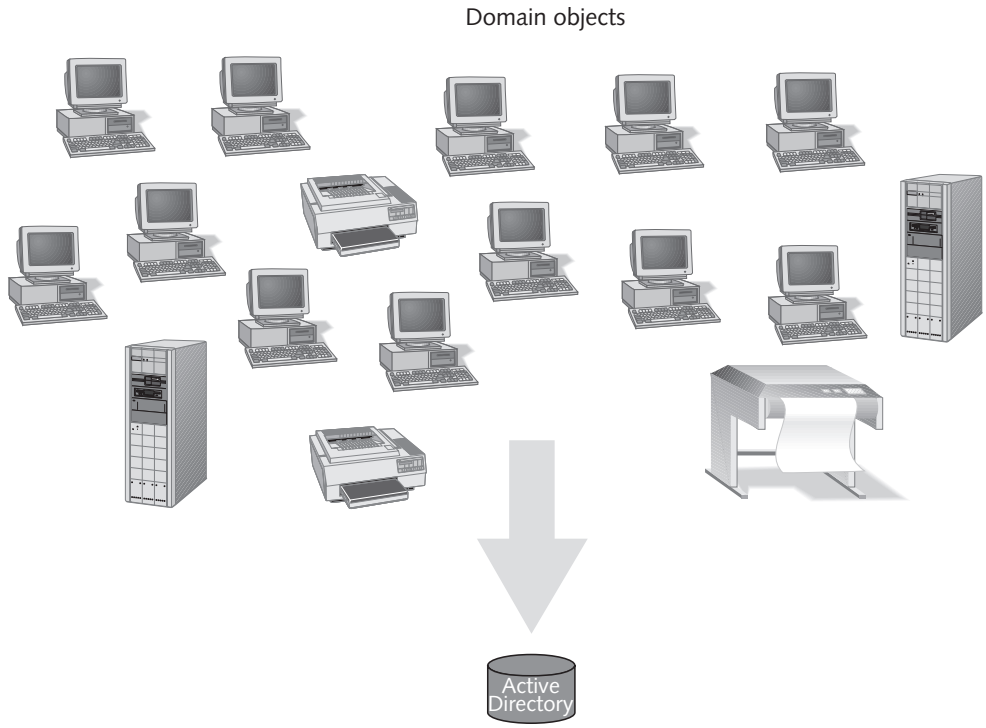

Figure 4-2 Domain objects in the Active Directory

In Windows 2000 Server, each DC is equal to every other DC in that it contains the full range of information that composes the Active Directory (see Figure 4-3). For example, when you create a new user account the information associated with that account can be created on any Windows 2000 Server DC, as opposed to the old Windows NT Server method of creating accounts only on the PDC. The Windows 2000 Server implementation of DCs is called **multimaster replication**. When an account is created, the full information about that account is replicated on every other DC. The advantage of this approach is that if one DC fails, the Active Directory is fully intact on all other DCs, and there is no visible network interruption, because you do not have to pause to manually promote a server to take over as the master database.

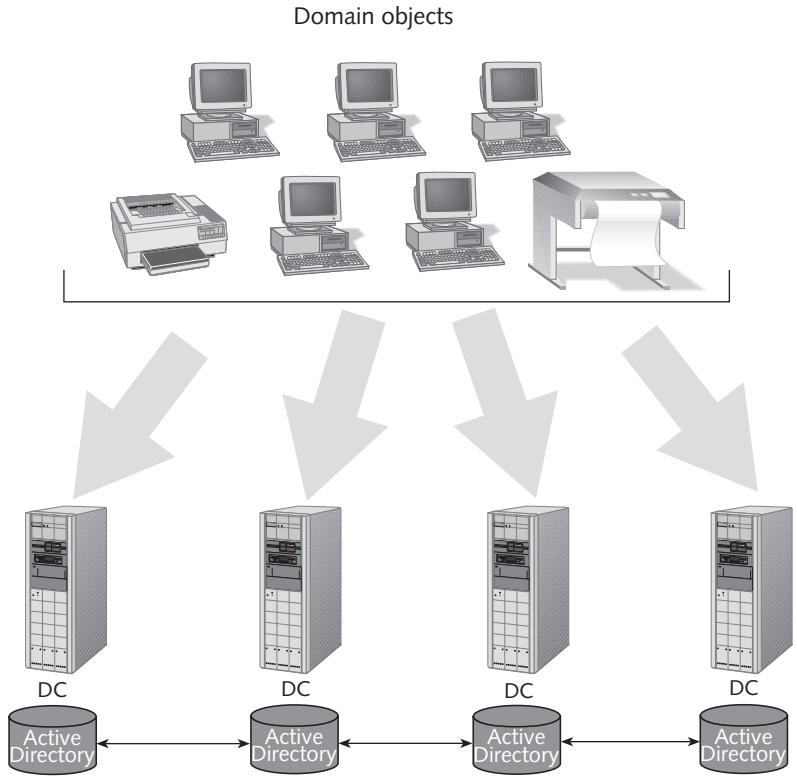

Figure 4-3 Windows 2000 Active Directory architecture

In Windows 2000 Server, you can set replication of Active Directory information to occur at a preset interval instead of as soon as an update occurs. Also, you can determine how much of the Active Directory is replicated each time it is copied from one DC to another.

In Windows NT 4.0, the process of replicating the PDC to one or more BDCs could create significant network traffic, particularly over a slow WAN link. This problem has been addressed in Windows 2000 Server in two ways: (1) Windows 2000 Server can replicate individual properties instead of entire accounts (as in Windows NT 4.0), which means that a single property can be changed without replicating information for the whole account, and (2) Windows 2000 Server can replicate the Active Directory on the basis of the speed of the network link, such as replicating more frequently over a LAN link than over a WAN link.

Two general concepts are important as a starting place for understanding the Active Directory: schema and namespace. These concepts are described in the next sections. (Try Hands-on Project 4-1 to practice installing the Active Directory.)

Schema

Each kind of object in the Active Directory is defined through a **schema**, which is like a small database of information associated with that object, including the object class and its attributes. To help you understand a schema, consider the characteristics associated with a vehicle. First, there are different classes of vehicles, including automobiles, trucks, tractors, and motorcycles. Further, each class has a set of attributes. For automobiles those attributes include engine, headlights, seats, steering wheel, dashboard, wheels, windshield, CD player, cup holder, and many others. Some of those attributes must be present in every automobile, such as an engine and wheels. Other attributes are optional—whether there is a CD player or cup holder, for instance.

A user account is one class of object in the Active Directory that is defined through schema elements unique to that class. The user account class as a whole has the following schema characteristics (see Figure 4-4):

- A unique object name
- A **globally unique identifier** (**GUID**), which is a unique number associated with the object name
- Required attributes (those that must be defined with each object)
- Optional attributes (those that are optionally defined)
- A syntax (format) to determine how attributes are defined
- Pointers to parent entities, such as to a parent domain

Examples of required user account attributes that must be defined for each account are:

- Username
- User's full name
- Password

Providing an account description or specifying if the account is enabled for remote access over a telephone line are examples of optional attributes that do not have to be completed when you create an account. In some instances, the attributes that are required and those that are optional can be influenced by the security policies that the server administrator sets in the Active Directory for a class of objects. This is true, for example, with account passwords because it is possible (but not recommended) for you to have a security policy that does not require account passwords.

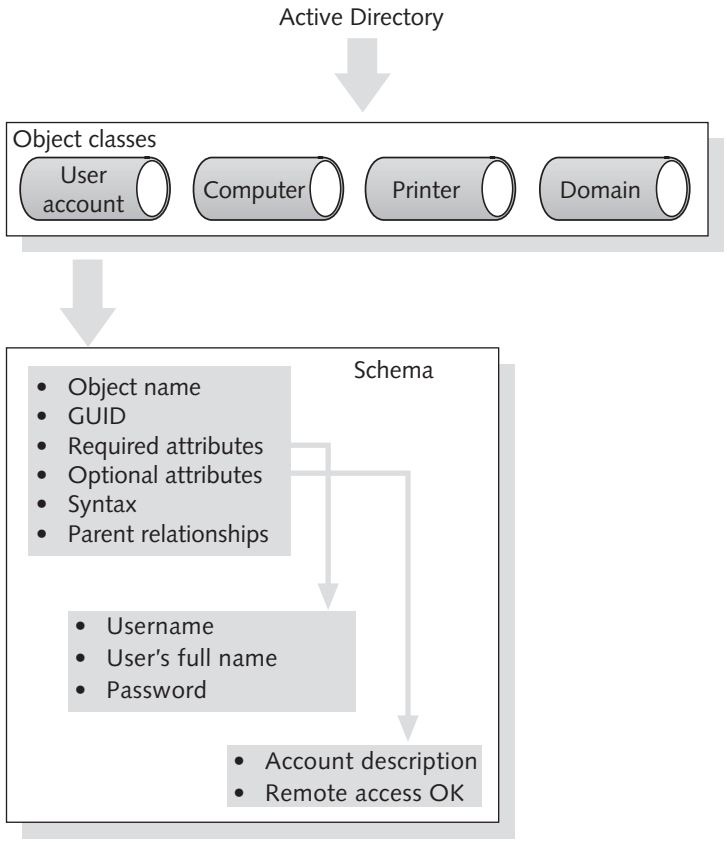

Figure 4-4 Sample schema information for user accounts

Each attribute is automatically given a version number and date when it is created or changed. This information enables the Active Directory to know when an attribute value, such as a password, is changed, and update only that value on all DCs.

When you install Windows 2000 Server for the first time on a network server, designating it as a domain controller, you also create several object classes automatically. The default object classes include:

- Domain
- User account
- Group
- Shared drive
- Shared folder
- Computer
- Printer

You can supplement the schema contents by using the Schema Manager snap-in for the Microsoft Management Console (MMC). Schema extensions can be added without rebooting Windows 2000 Server, which means that they can be used right after you add them.

Schema information for objects in a domain are replicated on every DC. Each object has a **common name** (**CN**) and a **distinguished name** (**DN**). The CN is the most basic and unique name for an object, which may be HPLaserMain for a printer, or the combination firstname and lastname for a user account, RobBrown for example. The DN of an object gives more information; it contains the name of the object, the object class name, and the name of any higher-level entities to which the object belongs, such as a domain. When an Active Directory client needs to access an object, such as a user account or printer, that client can use the DN for unmistakable identification of the object. In the example of RobBrown who belongs to the domain *tracksport.org*, the DN would be:

/DC=ORG/DC=tracksport/CN=Users/CN=RobBrown

Notice in this example, that there are two CNs—one is Users, which identifies the user account object class, and the other is RobBrown, which identifies one of the users within the Users class. Also, there are two DCs that equate to the namespace, *tracksport.org*.

This DN is also an example of a **relative distinguished name** (**RDN**), a DN in which part of the name is a reference to another part of the name. In this case, Users is a higher-level, or parent, object (accounts in a domain), and RobBrown is an attribute (one username) of that parent object. Both are linked in a two-way relationship. If you search for RobBrown as an account, your search route would follow the database structure from domain (tracksport.org), through Users, to RobBrown. If you are the user RobBrown and you want to change your password, you would go from account RobBrown through the parent object Users to update that information associated with the RobBrown account. The RDN relationship enables users to quickly find an object, such as a specific account, by going from the domain level to the accounts object level, and finally to the specific account. Consider how much longer it would take to perform a search for an account in a structure that did not divide objects into groupings, so that you would have to search every object (printers, domains, group names, accounts, etc.) until you found the single account you wanted to locate. This might be like trying to locate a city on a map that did not show states, provinces, countries, and continents.

Besides the DN, each object can be identified through its globally unique identifier (GUID), which is a hexadecimal number, such as 8112AF88BC42 to identify a specific computer, created by the Active Directory for an object. A GUID is never reused when you delete an object. You can, however, change the name of the object, but retain the same GUID, such as when you rename a user account.

Namespace

The Active Directory is also essential in providing Domain Name Service (DNS). As you learned in Chapter 3, DNS is a TCP/IP-based utility that converts dotted decimal addresses to computer and domain names and vice versa, through a process called **name resolution**. A computer running Windows 2000 Server can be set up to act as a DNS server on a network. For

example, when you send a TCP/IP request to connect with *microsoft.com*, the DNS server at Microsoft's site resolves that domain name to the address, 207.46.130.150. The DNS services are configured so that *microsoft.com* is a parent, or root, domain. Within that domain there are child objects, such as the Microsoft developers network (*msdn.microsoft.com*), which resolves to 207.46.130.161.

The ability to resolve names takes place in a designated logical area of a network, called a **namespace**, that is set up for this purpose. The namespace contains a domain name, such as *microsoft.com*. On a network consisting of Windows 2000 servers, namespace logic is composed of two key elements: (1) the Active Directory, which contains named objects and (2) one or more DNS servers that can resolve names. These services can be on a single computer, such as a Windows 2000 server in a small network that is set up as a DC and a DNS server. Or, they can be distributed across several servers on a large network, which might have two servers set up as DNS servers and 22 set up as DCs.

Microsoft recognizes two kinds of namespaces: contiguous and disjointed. A **contiguous namespace** is one in which every child object contains the name of the parent object, such as in the example of the child object *msdn.microsoft.com* and its parent object *microsoft.com*. When the child name does not resemble the name of its parent object, this is called a **disjointed namespace**, such as when the parent for a university is *uni.edu*, and a child is *bio.ethicsresearch.com*.

If you are an e-mail user, you may already be familiar with another type of name that enables you to quickly identify the account name and domain of someone with whom you communicate. That name is called the **user principle name (UPN)** and is in the format *username@domain*, such as *RobBrown@tracksport.org*.

Elements in the Active Directory

The Active Directory has a tree-like structure that is similar to the hierarchy of folders and subfolders in a directory structure. For example, in a directory structure information is stored in a root folder, which is at the highest level. The root folder may contain several main folders, 15 or 20, for instance. Under each folder there exist subfolders, and within subfolders there can be more subfolders. Subfolders can have a nearly infinite depth, but typically do not go more than five or ten layers deep. Just as files are the basic elements that are grouped in a hierarchy of folders and subfolders, objects are the basic elements of the Active Directory and are grouped in a hierarchy of larger containers. Also, just as the folder structure affects how you can set up security on a server, the Active Directory structure creates boundaries for security in a network enterprise. The hierarchical elements, or containers, of the Active Directory are the following (see Figure 4-5):

- Domains
- Organizational units (OUs)
- Trees
- Forests
- Sites

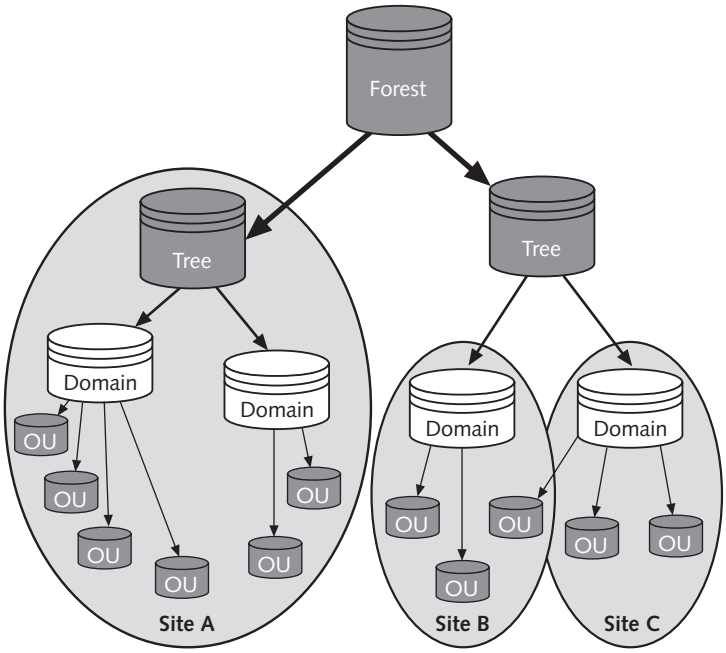

Figure 4-5 Active Directory hierarchical containers

Domain

A domain is a grouping of objects that typically exists as a primary container within the Active Directory. The basic functions of a domain are as follows:

- To provide a security boundary around objects that have a common relationship
- To establish a set of information to be replicated from one DC to another
- To expedite management of a set of objects

Domains can also be set up in Windows NT Server, but without the use of hierarchical containers or the Active Directory.

When you are migrating from Windows NT Server to Windows 2000 Server, an additional function of a Windows 2000 Server domain is to provide an easy migration path from an existing Windows NT Server domain.

When you use the server-based networking model (see Chapter 1) to verify users who log on to the network, there is at least one domain. For example, if you are planning the Active Directory for a small business of 34 employees, who have workstations connected to a network that has one or two Windows 2000 servers, then one domain is sufficient for that business (see Figure 4-6). The domain functions as a security boundary within which to group all of

the network resource objects consisting of servers, user accounts, shared printers, and shared folders and files. If there is Internet connectivity and one or more intranets, then the domain provides a security boundary to keep information within intranets secure from outside access via the Internet. The boundary establishes the capability to manage what information comes into the network from the Internet and what information goes out.

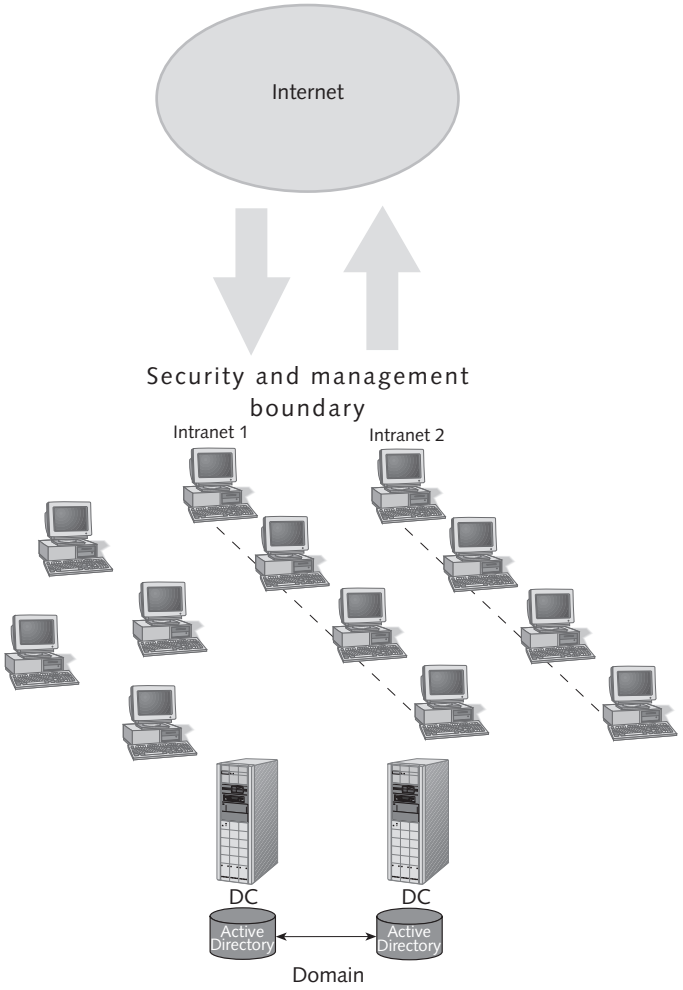

Figure 4-6 Single domain

In a medium or large business you might use more than one domain—for instance when business units are separated by long distances and you want to limit the amount of DC replication over expensive WAN links as well as establish tight security boundaries for each location. For example, consider a company that builds tractors in South Carolina and that has a parts manufacturing division in Japan. Each site has a large enterprise network of Windows 2000 servers,

and the sites are linked together in a WAN by an expensive satellite connection. When you calculate the cost of replicating DCs over the satellite WAN link, you cannot justify it in terms of the increased traffic that will delay other vital daily business communications. In this situation, it makes sense to create two separate domains, one for each site, as shown in Figure 4-7.

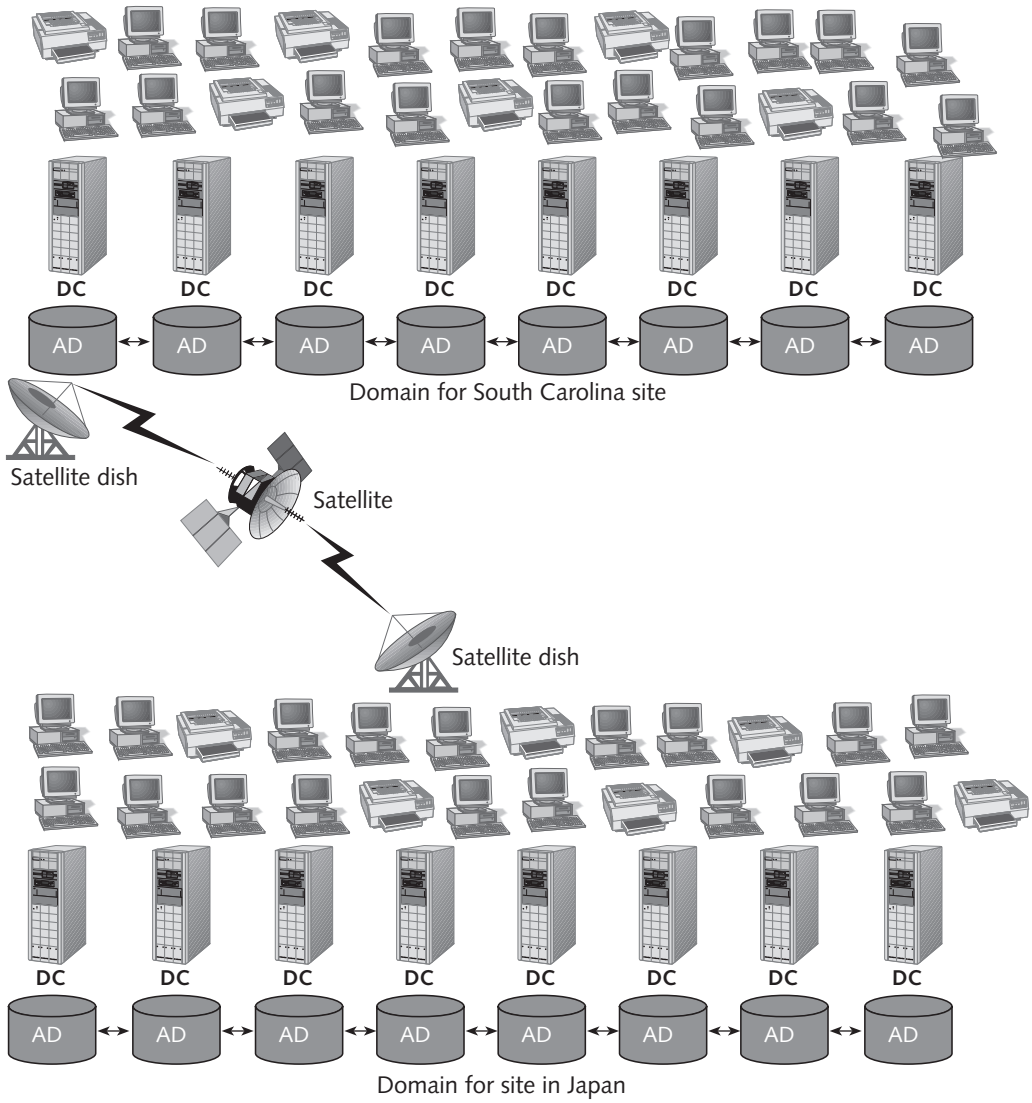

Figure 4-7 Using multiple domains

Microsoft has general guidelines for when to use a domain and when not to, as shown in Table 4-1, but keep in mind that when you set up the Active Directory there is at least one domain.

Table 4-1 Domain Creation Dos and Don'ts

Dos	Don'ts
Create a domain in circumstances that require special security measures between organizational groupings, such as departments, units, or divisions	Create domains that represent the organizational structure, because frequent reorganizations result in major restructuring of domains and the Active Directory
Create a domain for specialized management of particular resources (often also related to the security and network architecture)	Create domains along business process divisions, which are often political divisions within an organization, because new management may redefine business process activities, resulting in a major restructuring of domains and the Active Directory
Create a domain to migrate Windows NT servers to Windows 2000	
Create a domain when geography or WAN links make it difficult to replicate DCs between organizational groupings, such as departments, units, or divisions	

The guidelines in Table 4-1 are not set in stone, and you may find that a particular business process or unique set of requirements causes you to ignore them. For example, a research organization might have separate units, all of which work on different and highly classified projects. For the sake of security, each unit might be a separate domain, causing your domain structure to completely reflect the structure of the organization. As you plan in this type of situation, just remember that if there is a major restructuring of the organization, you will face equally major work in restructuring domains and resources in the Active Directory. For this reason, the Active Directory provides another alternative, called an organizational unit. (Try Hands-on Projects 4-2 and 4-3 to practice managing a domain and objects in a domain.)

Organizational Unit

An **organizational unit** (**OU**) offers a way to achieve more flexibility in managing the resources associated with a business unit, department, or division than is possible through domain administration alone. An OU is a grouping of related objects within a domain, similar to the idea of having subfolders within a folder. OUs can be used to reflect the structure of the organization without having to completely restructure the domain(s) when that structure changes.

OUs allow the grouping of objects so that they can be administered using the same group policies, such as security and desktop setup. OUs also make it possible for server administration to be delegated or decentralized. For example, in a software company in which the employees are divided into 15 project teams, the user accounts, shared files, shared printers, and other shared resources of each team can be defined as objects in separate OUs. There would be one domain for the entire company and 15 OUs within that domain, all defined in the Active Directory. With this arrangement, file and folder objects can be defined to specific

OUs for security, and the management of user accounts, account setup policies, and file and folder permissions can be delegated to each group leader (OU administrator).

Consider another example of a larger network and organization, which is a grocery chain that has three separate divisions: manufacturing, distribution, and retail. In this scenario, the manufacturing unit consists of five sites that are networked into a WAN, and the computer resources of that unit are managed by their own IT group of server administrators and programmers. The manufacturing unit provides prepared foods, which include canned items, frozen foods, bakery goods, soft drinks, and other foods. The distribution unit transports all food items to the retail stores and has its own independent IT group and network. Finally, the retail unit provides central management of hundreds of grocery stores throughout 20 states, and it networks each store into a central site through a WAN with computer resources managed by a third independent IT group. In this situation, there are three separate administrative units, each with its own IT group and unique management policies. Each administrative group can be incorporated into an individual OU, as in Figure 4-8.

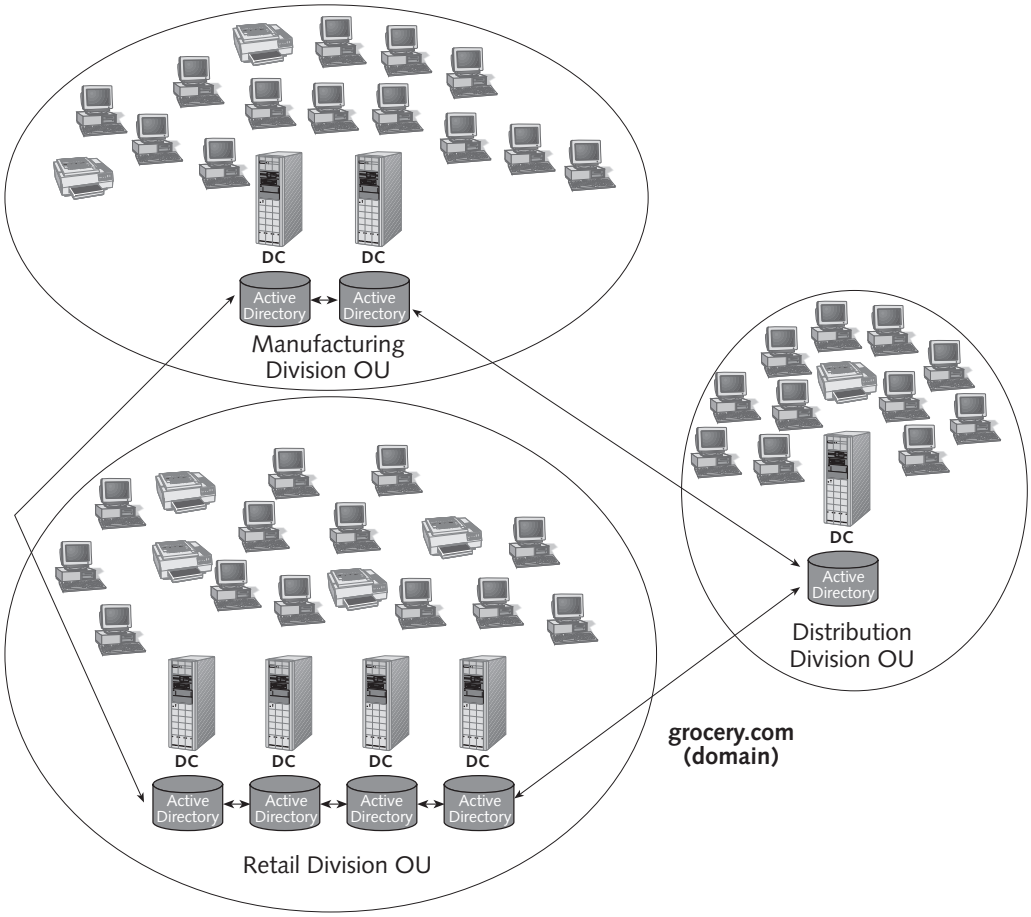

Figure 4-8 OUs used to reflect the divisional structure of a company

OUs can be nested within OUs, as subfolders are nested in subfolders, so that you can create them several layers deep. In the grocery chain example, you might have one OU under the Retail OU for the Accounting Department, an OU under the Accounting OU for the Accounts Receivable Group, and an OU under Accounts Receivable for the cashiers—creating four layers of OUs. The problem with this approach is that creating OUs many layers deep can get as confusing as creating subfolders several layers deep. It is confusing for the server administrator to track layered OUs, and it is laborious for the Active Directory to search through each layer.

When you plan to create OUs, keep three concerns in mind:

- Microsoft recommends that you limit OUs to 10 levels or fewer.
- The Active Directory works more efficiently (using less CPU resources) when OUs are set up horizontally instead of vertically. Using the grocery chain example, it is more efficient to create the Accounting, Accounts Receivable, and Cashier OUs directly under the Retail OU, resulting in two levels instead of four.
- The creation of OUs involves more processing resources because each request through an OU (for example to determine a permission on a folder) requires CPU time. When that request must go several layers deep through nested OUs, even more CPU time is needed.

Microsoft has several guidelines, which are presented in Table 4-2, to help you plan for OUs. Compare this table with Table 4-1 to assess when you might create an OU and when to create a domain.

Table 4-2 OU Creation Dos and Don'ts

Dos	Don'ts
Create OUs, as needed, to represent the organizational structure of departments, units, and divisions for different policies and to delegate administration	Create OUs more than 10 layers deep
Create OUs, as needed, to represent objects in the Active Directory that have similar policies, security, or other characteristics, such as shared printers or shared disk drives	Create more OUs than absolutely necessary
Create OUs, as needed, to represent specific project areas, such as for employees who are temporarily helping with the installation of a new client/server system	Create OUs for major security boundaries when this can be handled by a domain or by sites (discussed later), such as for IP traffic control
Create OUs, as needed, to represent the business process or political functions in an organization, such as an OU for the president's office, one for the Business Office, and one for each research group in a health research organization	Create OUs for DC replication

Tree

A **tree** contains one or more domains that are in a common relationship, and has the following characteristics:

- Domains are represented in a contiguous namespace and can be in a hierarchy.
- Two-way trust relationships exist between domains in which each domain can access the resources of the other.
- Member domains use the same schema for all types of common objects.
- Member domains use the same global catalog (a global catalog is something like an encyclopedia of information about objects and their attributes in all domains).

The domains in a tree typically have a hierarchical structure, such as a root domain at the top and other domains under the root. In the *tracksport.org* example, *tracksport.org* might be the root domain and have four domains under the root to form one tree: *east.tracksport.org*, *west.tracksport.org*, *north.tracksport.org*, and *south.tracksport.org,* as shown in Figure 4-9. These domains use the contiguous namespace format in that the child domains each contain the name of the parent domain.

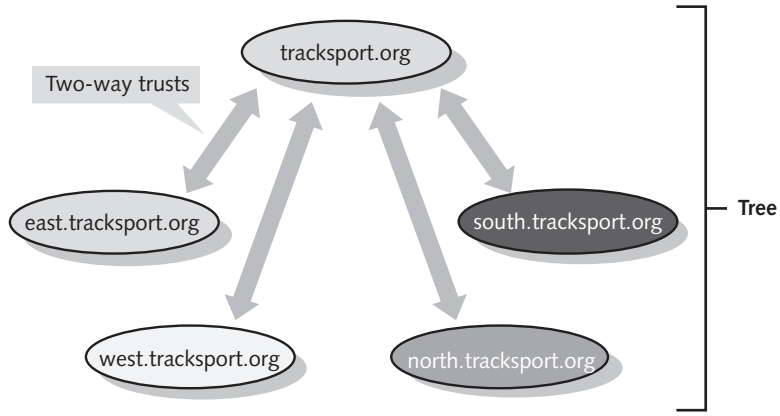

Figure 4-9 Tree with hierarchical domains

The domains within a tree are in what is called a Kerberos **transitive trust** relationship, which consists of two-way trusts between all domains (see Figure 4-9). This is similar to the universal trust relationship among domains in Windows NT Server. In a **two-way trust**, each domain is trusting and trusted. A **trusted domain** is one that is granted access to resources, whereas a **trusting domain** is the one granting access. In a two-way trust, members of each domain can have access to the resources of the other.

Because all domains have a two-way trust relationship, any one domain can have access to the resources of all others. The security in the two-way trust relationships is based on Kerberos techniques (see Chapter 1), using a combination of protocol-based and encryption-based security techniques between clients and servers. A new domain joining a tree has an

instant trust relationship with all other member domains, which makes all objects in the other domains available to the new one.

All domains in a tree share the same schema, which means that they share the same object classes and attributes. One important advantage of this arrangement is the way it affects security. For example, if the first domain in the tree requires password restrictions, such as a minimum password length, then all others will have the same restriction.

All member domains share the same global catalog. The **global catalog** is a subset of the Active Directory; it contains information about all objects in the domain where it resides and partial information (only selected schema elements) about objects on other domains. The first DC created in a domain is also set up by default as a global catalog server. The value of the global catalog is that DCs in one domain do not have to replicate their information to DCs in another domain. The global catalog serves the following purposes:

- Authenticating users when they log on
- Providing lookup and access to all resources in all domains
- Providing replication of key Active Directory elements
- Keeping a copy of the most used attributes for each object for quick access

Each tree must have at least one DC that is also configured to operate as a global catalog server. When you plan a tree, also plan the location of global catalog servers so that users are quickly authenticated for access in the tree. In the *tracksport.org* example, consider that each domain is separated by many miles, for example when the parent domain is in Washington, D.C., and the child domains are located in Boston, Los Angeles, Chicago, and Atlanta. In this situation, it makes sense to have a global catalog server in each location because authentication over WAN links is likely to be slow. If there is only one global catalog server in Washington, then users in the other four cities will have to wait longer to log on than users in Washington. If there is a global catalog server in each of the five cities, then logon response will be faster, and the WAN links will be free to give priority to other types of communications.

Table 4-3 lists the dos and don'ts for creating trees.

Table 4-3 Tree Creation Dos and Don'ts

Dos	Don'ts
Define main domains before defining a tree	Define a tree prior to creating the first domain
Plan the hierarchy of domains and use of OUs before creating a tree	Define a tree if you can use a single domain structure (a better alternative than using trees, if possible)
Define a tree when you have domains in different countries so that you can set up each domain to use a language native to the country where it resides	Define a tree if you must use a disjointed namespace

Table 4-3 Tree Creation Dos and Don'ts (continued)

Dos	Don'ts
Define a tree if you are planning multiple domains that will be administered at different sites by different people	
Create a tree and multiple domains when WAN connectivity is slow between distant sites, because global catalog replication transfers less information and requires less bandwidth than DC replication	

Forest

A **forest** consists of one or more trees that are in a common relationship and that have the following characteristics:

- The trees use a disjointed namespace
- All trees use the same schema
- All trees use the same global catalog

A forest provides a means to relate trees that use a contiguous namespace in domains within each tree but that have disjointed namespaces in relationship to each other. Consider, for example, an international automotive parts company that is really a conglomerate of separate companies, each having a different brand name. The parent company is PartsPlus located in Toronto. PartsPlus manufactures alternators, coils, and other electrical parts at plants in Toronto, Montreal, and Detroit and has a tree structure for domains that are part of *partsplus.com*. Another company that they own, Marty and Mike's (*2m.com*), makes radiators in two South Carolina cities, Florence and Greenville, and radiator fluid in Atlanta. A third member company, Chelos (*chelos.com*), makes engine parts and starters in Mexico City, Oaxaca, Monterrey, and Puebla, all in Mexico—and also has a manufacturing site in Valencia, Venezuela. In this situation, it makes sense to have a contiguous tree structure for each of the three related companies and to join the trees in a forest of disjointed name spaces, as in Figure 4-10.

The advantage of joining trees into a forest is that all domains share the same schema and global catalog. A schema is set up in the root domain, which is *partsplus.com* in our example, and the root domain is home to the master schema server. At least one DC functions as a global catalog server, but in our example, it is likely that you would plan to have a global catalog server located at each geographic site (domain).

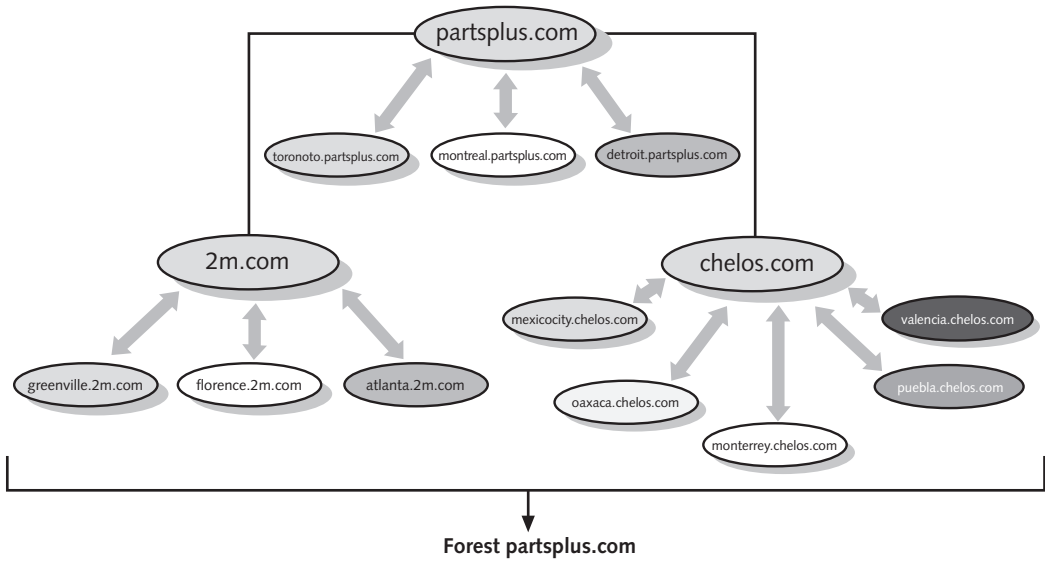

Figure 4-10 A forest

Kerberos trusts are transitive between trees in a forest, but they cannot be transitive between two or more forests. The Active Directory structure in Figure 4-10 is also called a **single forest** model by Microsoft. It is possible to join two or more forests for common communication in a model that Microsoft calls a **separate forest** (see Figure 4-11). In a separate forest, there cannot be transitive trusts between forests, which is a critical consideration when you plan the Active Directory. Establishing a separate forest means that replication cannot take place between forests, that there are different schema and different global catalogs, and that the forests cannot be blended into a single forest in the future. Table 4-4 lists guidelines for creating forests.

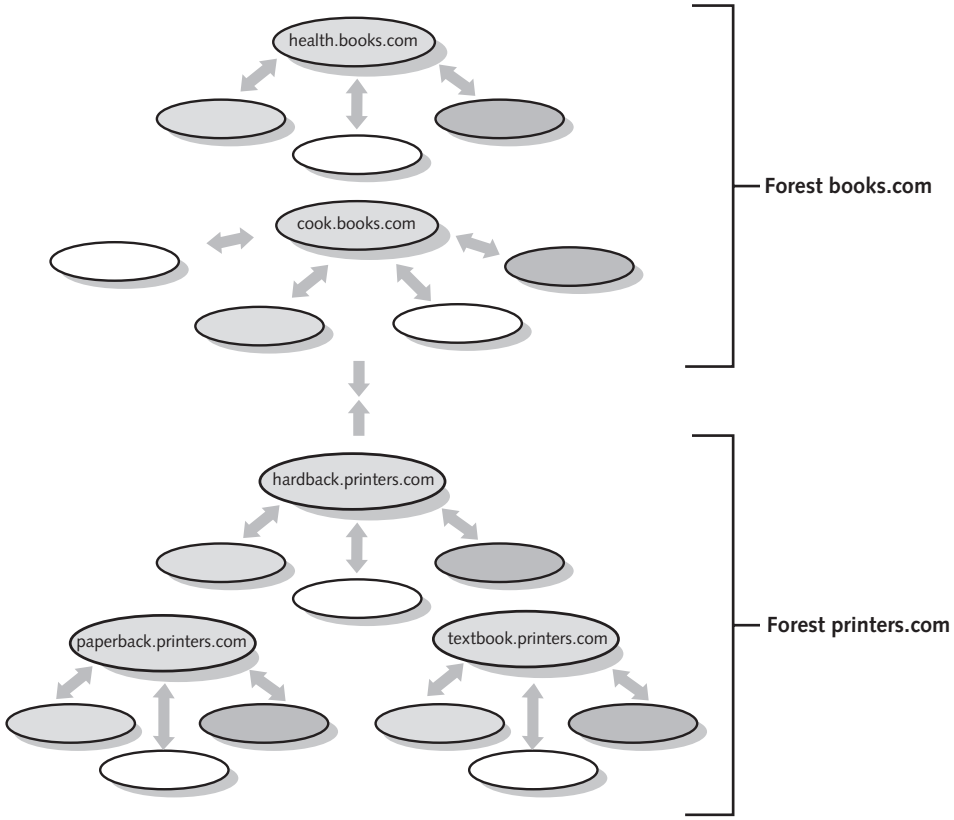

Figure 4-11 Separate forest model

Table 4-4 Forest Creation Dos and Don'ts

Dos	Don'ts
Create a forest to join trees/domains that can share schemas and global catalogs	Create forests when the member trees have little in common or cannot share the same schema
Create a single forest when there is no need to separate internal and external DNS resources between trees	Create a single or separate forest until you understand the security needs of all domains, trees, and potential forests
Create separate forests when the internal and external DNS resources must be keep separate between two or more forests	Create a separate forest when there is a possibility that the forests may merge into a single forest in the future
Establish a forest's name by using the name of the root domain or first domain in the first tree	Create a separate forest when the member forests must have a Kerberos transitive trust between them

Site

A **site** is a TCP/IP-based concept within the Active Directory that is linked to IP subnets and has the following functions:

- Reflects one or more interconnected subnets (see Chapter 3), usually connected at 512 Kbps or faster
- Reflects the same boundaries as the LAN it represents
- Is used for DC replication
- Is used to enable a client to access the DC that is physically closest
- Is composed of only two types of objects, servers, and configuration objects

Sites are based on connectivity and replication functions; therefore, they do not have a visible entry in the Active Directory or a namespace name. You might think of sites as a way of grouping Active Directory objects by physical location so the Active Directory can identify the fastest communications paths between clients and servers and between DCs. The physical representation of the network to the Active Directory is accomplished by defining subnets that are interconnected. For this reason, one site may be contained within a single OU or a single domain, or a site may span multiple OUs and domains, depending on how subnets are set up. The most typical boundary for a site consists of the LAN topology and subnet boundaries rather than the OU and domain boundaries.

There are two important reasons to define a site. First, by defining site locations based on IP subnets, you enable a client to access network servers using the most efficient physical route. In the PartsPlus example, it is faster for a client in Toronto to be authenticated by a Toronto global catalog server than for the client to go through Detroit or Mexico City. Second, DC replication is most efficient when the Active Directory has information about which DCs are in which locations.

Within a site, each DC replicates forest, tree, domain, and OU naming structures, configuration naming elements, such as computers and printers, and schema information. One advantage of creating a site is that it sets up redundant paths between DCs so that if one path is down, there is a second path that can be used for replication. This redundancy is in a logical ring format, which means that replication goes from DC to DC around a ring until each DC is replicated. If a DC is down along the main route, then the Active Directory uses site information to send replication information in the opposite direction around the ring. Whenever a new DC is added or an old one removed, the Active Directory reconfigures the ring to make sure there are two replication paths available from each DC.

The Windows 2000 Server feature that automatically adds or removes DCs is called the Knowledge Consistency Checker (KCC).

When information is replicated within a site, the information is sent from DC to DC uncompressed so there is less demand placed on each DC's CPU. Replication information

that is sent between different sites is compressed so that it goes over WAN links more quickly and creates less interference with other WAN communications. Intersite compression is in a ratio of up to 10 to1.

Two different sites are connected via a **site link object**, which is defined to the Active Directory. The schema of a site link object contains information that enables it to determine the cost of using different routes between sites. For example, there may be three network routes between two sites: one 1.2 Gbps ATM link, one 44.736 Mbps telecommunications link, and one 1.544 Mbps telecommunications link. A cost can be assigned to each link so that the Active Directory can determine their relative speeds and which route to use under which circumstances. The schema also contains information to show when one site link object connects to multiple sites, such as might be true for a satellite-based route. Last, there may be information to show when a particular link is available, because some links purchased over telecommunications lines may be available only during working hours, for example.

Two or more site link objects can be joined in a **site link bridge** object so that all member site links can communicate with one another. For example, if Site A is connected to Site B by site link object 1 and Site B is connected to Site C by site link object 2, then both site link objects can be combined by a site link bridge. The site link bridge enables communication between Site A and Site C. Normally, a site link bridge is really an IP router, as shown in Figure 4-12.

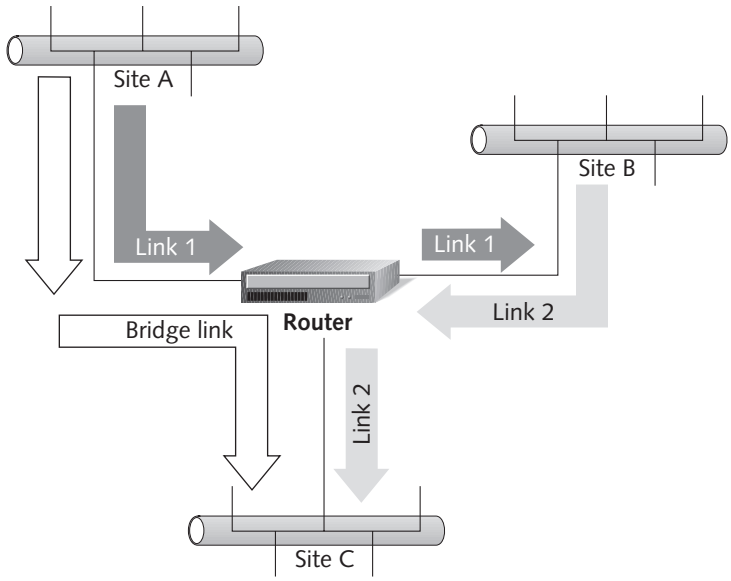

Figure 4-12 Site link bridge

Table 4-5 lists general site creation recommendations.

Table 4-5 Site Creation Dos and Don'ts

Dos	Don'ts
Create sites to reflect interconnected high-speed IP subnets	Create sites for small networks that have no IP subnets
Create sites on medium-sized and large networks to enable fast connectivity for users and for DCs	Create sites for IP links that have less than 128 Kbps of available bandwidth
Create additional sites on medium-sized and large networks when user connectivity and DC replication experiences slow response	Create extra sites to improve network performance without first determining what network congestion factors are causing poor performance
Create sites to enable ring-based DC fault tolerance	
Create one or more sites for a domain that encompasses two or more far-reaching geographic locations	

If you are new to IP communication, the concept of sites may at first be confusing. As you plan for a site keep in mind two factors. First, a site is simply a grouping of subnets and is really a concept that is independent of OUs, domains, trees, and forests (although one site might contain multiple OUs or domains). Second, the purpose of creating a site is to enable network traffic to go along the most efficient route on a medium-sized or large network. Whereas OUs, domains, trees, and forests are used to manage computer, printer, and user resources, sites are used to speed communication between resources.

Consider a state university's network that might take advantage of sites. The university has three domains, *students.uni.edu*, *faculty.uni.edu*, and *staffadmin.uni.edu*, organized into a single tree. Also, there are three campuses that are in different cities. The domains span each campus location. Thus *students.uni.edu* contains accounts and printers on DCs at all locations, for example. Each domain contains OUs that are appropriate to that domain, for instance *students.uni.edu* has an OU for students at each campus for a total of three OUs all at the same level. The campuses are relatively large with 7000 students, 10,000 students, and 18,000 students, and have networks that are physically divided into subnets. In this situation, you can designate each campus network as a site in the Active Directory, which enables it to find the fastest routes for traffic that is on-campus and for traffic that goes between campuses. For example, when a student logs on to *students.uni.edu*, the Active Directory can help that student find the nearest DC and avoid the chance that the logon authentication is performed over a WAN link at a different campus location. Another advantage is that the DC replication for each domain between sites (over WAN links) can be set to occur less frequently than replication within a site.

Active Directory Guidelines

The many components available in the Active Directory make its planning a potentially complex process. The following guidelines summarize the most important aspects of the Active Directory planning process covered so far in this chapter. Following them will simplify the process and help you to plan the best setup for your situation:

- Above all, keep the Active Directory as simple as possible and plan its structure *before* you implement it.
- Implement the least number of domains possible, with one domain being the ideal and building from there.
- Implement only one domain on most small networks.
- When you are planning for an organization that is likely to reorganize in the future, use OUs to reflect the organization's structure.
- Create OUs horizontally and not vertically within a domain.
- Create only the number of OUs that are absolutely necessary.
- Do not build an Active Directory with more than 10 levels of OUs (and hopefully no more than one or two levels).
- Use domains for natural security boundaries.
- Implement trees and forests only as necessary.
- Use trees for domains that have a contiguous namespace.
- Use forests for multiple trees that have disjointed namespaces between them.
- Use sites in situations where there are multiple IP subnets and multiple geographic locations, as a means to improve logon and DC replication performance.

Security Basics

Windows 2000 Server has several levels of security, which include the following:

- Account or interactive logon security
- Object security
- Services security (network authentication)

Account logon security involves making sure that each computer that accesses network servers has authorization through a preestablished account. Object security includes providing a list of which accounts can access a particular object, such as a shared folder or printer, and what type of access is permitted for each account. Network services security is determined by providing access to specific accounts and defining the extent of that access. Each of these options is discussed in the following sections.

Interactive Logon Security

Whenever a user accesses one or more Windows 2000 servers, he or she logs on to an account that is defined on a domain controller (DC) and is part of the Active Directory information. The DC checks to make certain that the user account is already defined and also authenticates the logon by checking the exact account name and password that the user provides from his or her workstation.

It is possible to set up a Windows 2000 server on a small network to act as a local computer that does not run the Active Directory service. In this case, the Windows 2000 server acts as a simple server that authenticates accounts only on that server. This use of Windows 2000 Server makes the computer act as just another local computer on the network, similar to a Windows 2000 Professional computer, but capable of handling far more accounts. This use of Windows 2000 Server is not recommended unless you have a small network of perhaps 10 to 20 users, and you have minimum security requirements and no connections to other networks.

Windows 2000 Server performs authentication using three approaches.

1. The default authentication is through Kerberos using a password or a **smart card**, a card about the size of a credit card that contains access information and can be plugged into a computer. To use this method, you must set up each Windows 2000 DC to authenticate via Kerberos and each workstation must also be set up as a Kerberos client.
2. Another option is to use Windows NT LAN Manager, which is an earlier method of authentication used by Windows NT servers and their clients. This method is used on a network that contains both Windows 2000 servers and Windows NT servers or when Windows NT server domain authenticates to a domain that has Windows 2000 servers.
3. A third authentication method is **Secure Socket Layer/Transport Layer Security** (**SSL/TLS**), which is used to authenticate a secure Windows 2000 Web server, for example. SSL/TLS uses **certificates** to authenticate a connection; the certificate is an encrypted set of information associated with a workstation, equivalent to a unique digital fingerprint. When a workstation requests access to the Web server, it sends its certificate, and the server responds by sending back a certificate to complete the authentication. To use this kind of authentication, you must first enable the **Extensible Authentication Protocol** (**EAP**) at the Web server.

Object Security

Each object in the Active Directory has a set of **security descriptors** that define how that object may be accessed. For example, a server will have a set of accounts and information about the type of access each account is allowed. Another example is a folder that has security descriptors to show which users can access that folder and what type of access they are allowed. A set of security descriptors is called an **access control list** (**ACL**), and it contains

all information about access to a particular object. A shared folder on a server called Payroll with an ACL that specifies only the accounts RBrown, LMason, AGonzales, and MKlein can have full access in an organization of 275 employees, while another folder called Paypolicies has an ACL that includes read-only access for everyone in the organization.

Each ACL for an object typically contains three categories of information:

- The user accounts (or account groups) that can access the object
- The permissions that determine the type of access
- The ownership of the object (the default owner of an object is its creator; ownership can be taken by another user account if that account has sufficient permission)

Each user account or group of accounts is assigned a type of access to an object, called a **permission**. There are standard permissions and special permissions. A standard permission is most frequently used and consists of the object permissions that are available by default. The types of permissions available are related to the nature of the object and appropriate security that applies to the object. The typical standard permissions that are available for objects in the Active Directory are as follows:

- *Deny*: No permission to access the object or a restriction on certain types of access
- *Read*: Permission for viewing an object, reading its contents (for folders and files), and determining properties or attributes of the object
- *Write*: Permission to change object properties or the contents of an object (for folders and files)
- *Delete All Child Objects*: Permission to remove an object, for example from an OU or domain
- *Create All Child Objects*: Permission to add an object, such as to an OU or domain
- *Full Control*: Permission to access the object for nearly any purpose, including to take ownership of the object or to change the permissions associated with that object

A special permission is used in situations in which a standard permission must be more finely tuned for a particular kind of access. For example, you may need to give applications developers in the organization Full Control to a folder but want to modify access so that some elements of Full Control are not available, such as the ability to change permissions. Figure 4-13 shows how you can set special permissions for the folder WINNT (see Chapter 9 for information about how to access and configure this dialog box).

When you belong to one user group that has limited permissions, such as Read, and one group that has more permissions, such as Full Control, then your access defaults to the highest level of permissions (Full Control in this example). The exception to the rule is Deny, which supercedes other permissions. For instance, if you belong to a group that has Deny Access to an object, and another group that has Full Control, then you have no access to that object. The exception to the last example is the group with Administrator privileges, because administrators must always have access to manage all server resources.

All objects have an owner who is by default the user account that created the object. The object owner has Full Control permission when the object is first created. An owner cannot transfer ownership to another account, but instead the new owner must take ownership, which means that the new owner must first have the right permission level to take ownership.

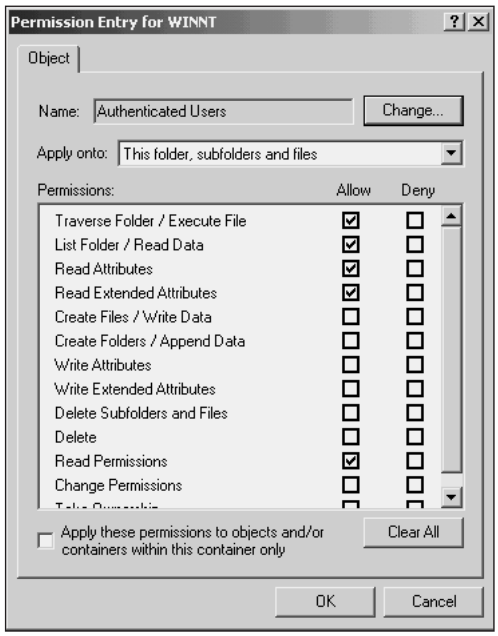

Figure 4-13 Special permissions for a folder

Services Security

Access to services offered by Windows 2000 Server can be controlled through a security feature called network authentication. A Windows 2000 server may offer many different kinds of services to clients, such as DHCP or WINS (see Chapter 3). When you configure one of these services, you can specify which users and groups have read access to enable them to view and use the service (Figure 4-14). Another example is the ability to run services that perform specific tasks on a Windows 2000 Server, which include the logon service, the backup service,

and others. These abilities are **rights** and are usually associated with a group of users, such as administrators, backup operators, or all users. The security that is available for a service depends on the type of service and its purpose. When you install a service (see Chapter 6), make sure that you plan in advance who should have access to it and what type of access is needed.

Only give all users default access to an object or a service when it is installed. In most situations this type of access is inappropriate. You should always check the security, make the necessary adjustments, and test the security before you make an object or service available network-wide following its installation.

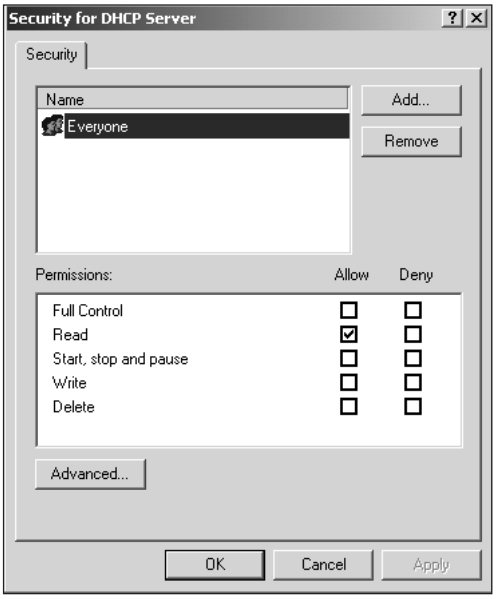

Figure 4-14 DHCP security

Using Groups

Windows 2000 Server employs two general types of groups, those used for security and those used for e-mail lists. **Security groups** are groups of users that are created as a way of reducing the amount of work you have to do when administering security. For example, if you are managing an Active Directory in which there are 700 users, it is much easier to divide the users into logical groupings on the basis of the type of security they will need and then assign security to each group. It is much faster to assign security to 10 or 15 groups than to assign security individually to 700 users. This principle still holds true even if you have far fewer users, so that you can spend more of your time on tasks other than setting up security.

Security groups appear in ACLs, but can also be used as e-mail distribution lists, which are lists of users created so that an e-mail message can be sent one time to all users on the list. Figure 4-15 shows the members of the Windows 2000 Server security group called DHCP Administrators, who have access to manage the DHCP server service.

The other type of group is called a **distribution group** and is intended strictly for use as an e-mail distribution list, for example for Microsoft Exchange. Distribution groups do not appear in ACLs.

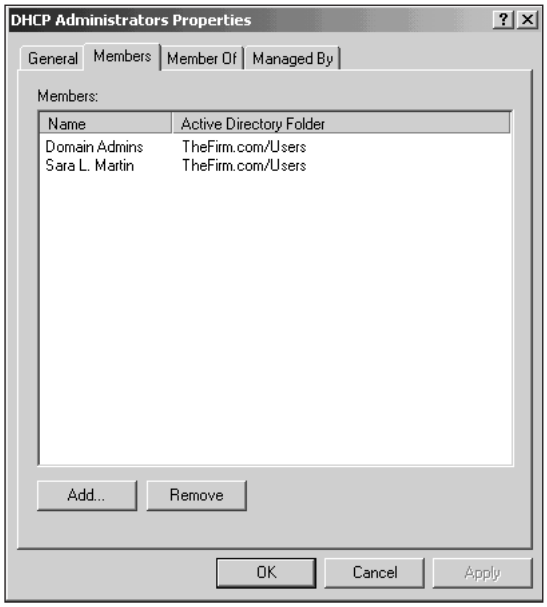

Figure 4-15 DHCP Administrators security group

Group Policies

When sites, domains, OUs, and groups are created, it is possible to establish group policies that apply to those Active Directory elements. For example, in your organization there might be a need to limit access to network resources in a specific OU by limiting what options appear on the desktop of the client workstations, such as eliminating the My Network Places icon in Windows 2000 Professional. Network access by these users might be limited to a menu that automatically appears when they log on to the network. In another situation, you might decide to eliminate a group's access to the Control Panel or you might remove the Run option from the Start button. Group policies are set up by installing the Group Policy snap-in in the Microsoft Management Console (MMC).

Security Templates

In Windows NT 4.0 and earlier, security that applies to groups is set by using several different tools, such as User Manager for Domains, Event Viewer, My Computer, and others.

Windows 2000 Server manages security policies from one location, the Security Templates snap-in for the MMC (Figure 4-16, in which the MMC setup has previously been saved as the file, security.msc). This snap-in enables you to set up security that governs the following (try Hands-on Projects 4-4 and 4-5):

- Account policies
- Local server or domain policies
- Event log tracking policies
- Group restrictions
- Service access security
- Registry security
- File system security

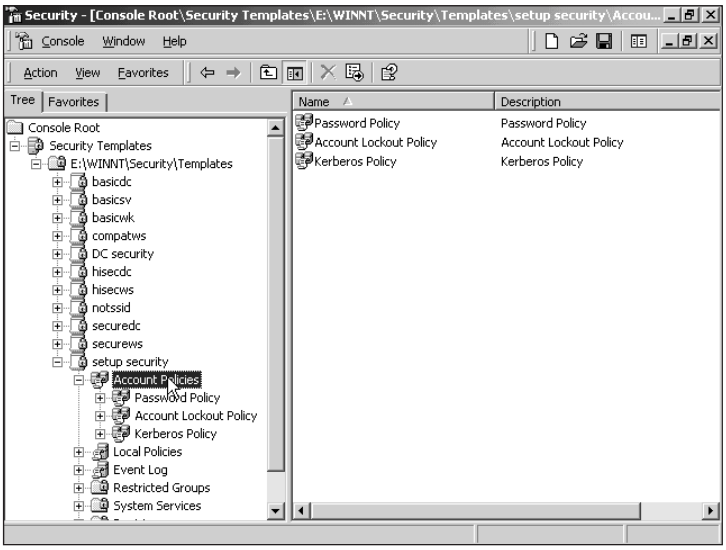

Figure 4-16 Security Templates snap-in

Before you create a security template, develop a plan to match your security needs. Your plan should address the following questions:

- What password restrictions should apply to accounts?
- What security restrictions are needed for folders and files, and how should those restrictions be inherited by subfolders within folders?
- What elements of the Registry should be secured?
- What restrictions should apply to individual server services, such as access to the Event logging service or to printing services?
- Should account activity be tracked through auditing?
- Which accounts should be allowed to log on to the server locally and which should be able to log on through the network?
- Which accounts should be allowed to schedule tasks to automatically run on the server?

When you are ready to create a security template, use these general steps:

1. Make sure there is no default security template that already matches what you want to do.
2. Make sure that the Group Policy and Security Templates snap-ins are installed in the MMC.
3. Create a security template by clicking the main folder under Security Templates in the MMC (such as \WINNT\Security\Templates), Action, and New Template.
4. Enter a name for the template and a description, and then click OK.
5. Double-click the new template in the right pane and configure the appropriate elements, such as System Services.
6. Import your newly created template to an existing group policy by installing the Security Configuration and Analysis snap-in, right-click that snap-in, open a database or create a new one, double-click the template configuration file (.inf f.6) you created (only if you created a new database), right-click the Security Configuration and Analysis snap-in again, and then click Configure Computer Now.

You can change security directly for a specific Active Directory grouping, such as an OU, by modifying the group policy settings in the properties for for that OU. However, it is more effective planning to create a security template for different Active Directory groupings and then import the template. For example, if you have 20 OUs set up in a domain and wanted to use one security policy for 10 OUs and a different one for the other 10 OUs, then you would create two security templates and import the appropriate template for each OU to the MMC Group Policy snap-in.

IP Security Policies

Windows 2000 supports the implementation of **IP security** (**IPSec**), which is a set of IP-based secure communications and encryption standards created through the Internet Engineering Task Force (IETF). When an IPSec communication begins between two computers, the computers first exchange certificates to authenticate the receiver and sender. Next, data is encrypted at the NIC of the sending computer as it is formatted into an IP packet which consists of a header containing transmission control information, the actual data, and a footer with error-correction information (see Chapter 3). IPSec can provide security for all TCP/IP-based application and communications protocols, including FTP and HTTP, which are used in Internet transmissions (see Chapter 3). IPSec policies are managed through the IP Security Policy Management snap-in in the MMC. A computer that is configured to use IPSec communication can function in any of three roles:

- *Client (Respond Only)*: When Windows 2000 Server is contacted by a client using IPSec, it will respond by using IPSec communication. This mode is also called responder.

- *Server (Request Security)*: When Windows 2000 Server is first contacted or when it initiates a communication, it will use IPSec by default. If the responding client does not support IPSec, Windows 2000 Server will switch to the clear mode, which does not employ IPSec. This role is also called the initiator.
- *Secure Server (Require Security)*: Also called the lockdown role; Windows 2000 Server will only respond using IPSec communication, which means that communication via any account and with any client is secured through strict IPSec enforcement.

When you use the Secure Server setup and also plan to set up Windows 2000 Server for SNMP communication, for example to monitor the network through the Windows 2000 Network Monitoring tool, you should consider omitting SNMP communication from IPSec. You can do this by establishing a filter in the IP Security Policy Management snap-in. The advantage of omitting SNMP communication is that you can still monitor activity from non-IPSec capable computers in order to track network problems.

IPSec security policies can be established through the IP Security Policies snap-in so that specific security standards apply to all computers that log on to a domain in the Active Directory. When you right-click IP Security Policies in the MMC and click Create IP Security Policy, Windows 2000 Server starts the IP Security Policy Wizard (see Figure 4-17) to help guide you through the steps in creating a security policy. A security policy consists of your specifications for what security methods to use for client and server communication, what IP filters to apply to communications, and which domain or domains are affected by the policy.

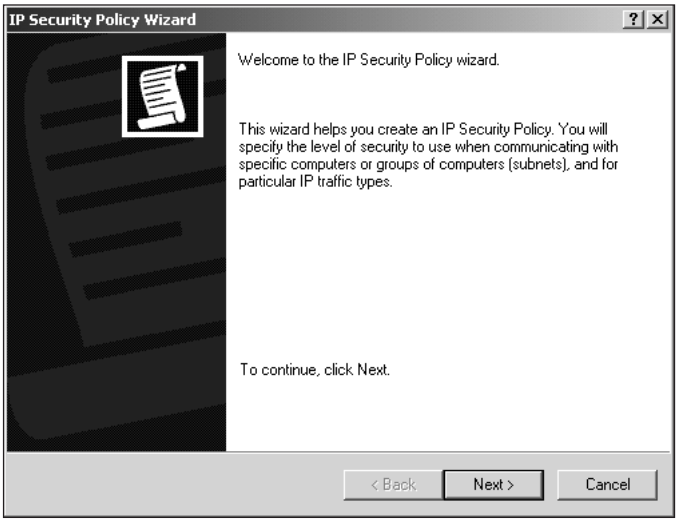

Figure 4-17 IP Security Policy Wizard

Chapter Summary

- The Active Directory and security are closely related in Windows 2000 server, and the best step you can take before setting up a server is to plan these elements thoroughly. The Active Directory is a set of directory services that enable you to manage a server, multiple servers, and an enterprise network. The most basic component of the Active Directory is an object. Each object is defined through an information set called a schema.
- In most cases, you will define one or more domains in the Active Directory, consisting of user accounts, printers, and other network resources. Because domains are difficult to restructure, for management purposes you have the option to divide a domain into smaller containers called organizational units, which can reflect your organization's structure. For large organizations, domains can be organized into trees and multiple trees, can be organized into forests. Keep in mind that an Active Directory can quickly become complex as you populate it with combinations of organizational units, domains, trees, and forests. Often the best rule when implementing the Active Directory is to keep it as simple and manageable as possible.
- For organizations that use TCP/IP communication and subnets, the Active Directory offers the ability to configure sites, which are often related to geographic locations or clusters of subnets. Sites enable you to configure physical network communications components for the best performance on a medium-sized to large network, taking advantage of existing subnets.
- By grouping objects in containers, such as OUs and domains, you have the ability to control and customize security in Windows 2000. Security can be attached to user accounts and other objects as well as to services. For example, you can require that all users in a domain have a password that is at least eight characters long. Some of the tools that you can use to configure security for Active Directory containers include group policies, security templates, and IPSec.

In the next chapter, you begin to put your advance planning to work as you learn how to install Windows 2000 Server. You find that the installation process is relatively simple because of your preparations through planning.

Key Terms

access control list (**ACL**) — A list of all security descriptors that have been set up for a particular object, such as for a shared folder or a shared printer.

certificate — An encrypted set of information associated with a workstation that is equivalent to a unique digital fingerprint and that is used to authenticate logon to a server, such as a Web server.

common name (**CN**) — The most basic name of an object in the Active Directory, such as the name of a printer.

contiguous namespace — A namespace in which every child object contains the name of its parent object.

disjointed namespace — A namespace in which the child object name does not resemble the name of its parent object.

distinguished name (**DN**) — A name in the Active Directory that contains all hierarchical components of an object, such as that object's organizational unit and domain, in addition to the object's common name. The distinguished name is used by an Active Directory client to access a particular object, such as a printer.

distribution group — A list of Windows 2000 Server users that enables one e-mail message to be sent to all users on the list. A distribution group is not used for security and thus cannot appear in an ACL.

domain — A grouping of resource objects, for example, servers and user accounts, that is one element of the Active Directory in Windows 2000 Server. A domain usually is a higher-level representation of how a business, government, or school is organized, for example reflecting a geographical site or major division of that organization.

domain controller (**DC**) — A Windows 2000 server that contains a full copy of the Active Directory information, that is used to add a new object to the Active Directory, and that replicates all changes made to it so those changes are updated on every DC in the same domain.

Extensible Authentication Protocol (**EAP**) — A protocol used to provide a range of security services for different manufacturer' s security devices, such as smart cards. EAP is used with other remote access protocols, for example for security through the Internet.

forest — A grouping of trees that each have contiguous namespaces within their own domain structure, but that have disjointed namespaces between trees. The trees and their domains use the same schema and global catalog.

global catalog — A grand repository for all objects and the most frequently used attributes for each object in all domains. Each tree has one global catalog.

globally unique identifier (**GUID**) — A unique number, up to 16 characters long, that is associated with an Active Directory object.

IP security (**IPSec**) — A set of IP-based secure communications and encryption standards created through the Internet Engineering Task Force (IETF).

Kerberos transitive trust relationship — A set of two-way trusts between two or more domains in which Kerberos security is used.

multimaster replication — In Windows 2000 Server, there can be multiple servers, called DCs that store the Active Directory and replicate it to each other. Because each DC acts as a master, replication does not stop when one is down, and updates to the Active Directory continue, for example creating a new account.

name resolution — A process used to translate a computer's domain name into the object that it represents, such as to a dotted decimal address associated with a computer, and vice versa.

namespace — A logical area on a network that contains directory services and named objects, and that has the ability to perform name resolution.

object — A network resource, such as a server or a user account, which has distinct attributes or properties, which is usually defined to a domain, and which exists in the Windows 2000 Active Directory.

organizational unit (**OU**) — A grouping of objects, usually within a domain, that provides a means to establish specific policies for governing those objects and that enables object management to be delegated.

permission — In Windows 2000, privilege to access an object, such as to view the object or to change it.

relative distinguished name (**RDN**) — An object name in the Active Directory that has two or more related components, such as the RDN of a user account name that consists of User and the first and last name of the actual user.

right — In Windows 2000, access privileges for high-level activities such as logging on to a server from the network, shutting down a server, and logging on locally.

schema — Elements used in the definition of each object contained in the Active Directory, including the object class and its attributes.

Secure Sockets Layer/Transport Layer Security (**SSL/TLS**) — An authentication method that uses certificates to verify users' right to access a remote server, such as a Web server.

security descriptor — An individual security property associated with a Windows 2000 Server object, for example to enable the account MGardner (the security descriptor) to access the folder, Databases.

security group — A group of Windows 2000 Server users that assign access privileges to objects and services. Security groups appear in ACLs.

separate forest — An Active Directory model that links two or more forests in a partnership; however, the forests cannot have Kerberos transitive trusts or use the same schema.

single forest — An Active Directory model in which there is only one forest, with interconnected trees and domains that use the same schema and global catalog.

site — An option in the Active Directory to interconnect IP subnets so that the server can determine the fastest route to connect clients for authentication and to connect DCs for replication of the Active Directory. Site information also enables the Active Directory to create redundant routes for DC replication.

site link bridge — An Active Directory object that combines individual site link objects to create faster routes, when there are three or more site links.

site link object — An object created in the Active Directory to indicate one or more physical links between two different sites.

smart card — A security device that contains information such as access keys, passwords, and a personal identification number (PIN). The smart card is about the size of a credit card and can be plugged into a computer.

transitive trust — A trust relationship between two or more domains in a tree in which each domain has access to objects in the others.

tree — Related domains that use a contiguous namespace, share the same schema, and have two-way, transitive trust relationships.

trusted domain — A domain that has been granted security access to resources in another domain.

trusting domain — A domain that allows another domain security access to its resources and objects, such as servers.

two-way trust — A domain relationship in which both domains are trusted and trusting, enabling one to have access to objects in the other.

user principle name (UPN) — A name that combines an account name with the domain name, such as *RobBrown@tracksport.org*, for easy identification, such as in e-mail.

Review Questions

1. You are in a meeting to plan the Active Directory at a college, and a colleague suggests that your organization create a top-level OU for all academic departments and then sequentially create one OU under the next for each of the 22 academic departments so that the OUs are 23 layers deep. Which of the following best matches your response?
   a. You endorse the plan.
   b. You recommend using 23 trees instead of OUs.
   c. You recommend creating the top-level department's OU and then creating the other 22 OUs directly under it so that the OUs are only two layers deep.
   d. You endorse the plan, but add that an OU should be created for each faculty member.
2. Members of the business division in your organization have the ability to view the contents of files in a shared server folder called Vendors and to create new files. This is an example of a
   a. permission.
   b. right.
   c. trust.
   d. transitive trust.
3. Your organization has a root domain called *consult.com* and other child domains called *web.cs.com* and *products.conslt.com*. This is an example of a
   a. disjointed namespace.
   b. contiguous namespace.
   c. commercial namespace.
   d. nonreciprocal namespace.
4. Which of the following would most likely be required in the schema for a user account?
   a. username
   b. user's full name
   c. user's room number or address
   d. all of the above
   e. only a and b
   f. only b and c

5. When an IPSec communication begins, it starts by
   a. encrypting data.
   b. checking the end-to-end network route for intruders.
   c. sending certificates.
   d. verifying the client's operating system license.
6. When a domain joins a tree,
   a. it is trusted but not trusting to other domains in the tree.
   b. it immediately forms a transitive trust relationship with other domains in the tree.
   c. it has a trust relationship only with the OUs under it.
   d. it gains Full Control permission of all objects in the other domains within that tree.
7. You are planning to migrate objects in a Windows NT 4.0 Server domain to a new set of Windows 2000 servers and then retire Windows NT 4.0. How might you prepare for the migration?
   a. First, create a tree in the Windows 2000 Active Directory.
   b. Create OUs in the Windows 2000 Active Directory, and migrate the Windows NT domain to each OU.
   c. Designate a distribution group in the Windows 2000 Active Directory to first test the migration.
   d. Create an equivalent domain in the Windows 2000 Active Directory, and migrate the Windows NT domain to the Windows 2000 domain.
8. What is the minimum number of domain controllers (DCs) that must exist in a domain?
   a. 1
   b. 2
   c. 3
   d. one for each OU
9. A global catalog server is also a
   a. router.
   b. WAN link.
   c. domain controller.
   d. DNS server.
10. Your boss has won a NASA contract to develop a new satellite guidance system that must be kept top-secret. Which of the following IP security measures would you recommend in planning for the new server purchased for this project?
    a. Client (Respond Only)
    b. Secure Server (Require Security)
    c. Server (Request Security)
    d. Certificate (Responder Security)

11. A group of 25 physicians is implementing a new network that uses two Windows 2000 servers and that will have Internet access. Besides the physicians, there is 1 bookkeeper, a business manager, 5 billing clerks, 28 nurses, and 5 nurse practitioners. Each will have a connection to the network. How many domains are needed when the Active Directory is configured?
    a. 1
    b. 2, one for access within the building, and one for physicians who dial in remotely from home
    c. 3, one for the physicians, one for the nurse practitioners and nurses, and one for the remaining staff
    d. 4, one for the physicians, one for the nurse practitioners, one for the nurses, and one for the remaining staff
12. You have just created a new security template. What is the last step you should perform in the process?
    a. Turn off sharing violation so that the template does not interfere with other templates that have similar names.
    b. Import the template to a group policy.
    c. Delete the existing group policy for the domain so that the template can go into effect.
    d. Remove the Security Templates snap-in from the MMC.
13. In a forest, all trees
    a. use the same global catalog.
    b. use the same schema.
    c. use a disjointed namespace between trees.
    d. all of the above
    e. only a and c
    f. only b and c
14. A set of security descriptors associated with an object compose that object's
    a. unique ID.
    b. access control list.
    c. schema.
    d. security template.
15. The employees in the research group of your organization have decided to use the same desktop setup so that it is easy to go from one computer to another in the labs. Which tool enables you to provide this service for them?
    a. User Manager
    b. Computer Manager
    c. Group Policy snap-in
    d. Security Configuration snap-in

16. Which of the following are objects in the Active Directory?
    a. shared printers
    b. domains
    c. shared folders
    d. all of the above
    e. only a and c
    f. only a and b
17. Your community college has a main campus and two large branch locations in a city of over 2 million persons. The main campus and branch locations consist of routed networks that have multiple subnets. What Active Directory component could make DC replication over the LAN and WAN links most efficient?
    a. setting up sites
    b. setting up domains
    c. setting up OUs
    d. setting up trees
18. When a DNS server converts a computer name to a dotted decimal address, this is called
    a. name recognition.
    b. name resolution.
    c. pinging.
    d. piping.
19. Which OU structure in the Active Directory is likely to result in the most CPU resource use?
    a. OUs nested 5 layers deep
    b. OUs nested 15 layers deep
    c. 5 OUs on the same level
    d. 15 OUs on the same level
20. A site link bridge is most likely to be a(n)
    a. Active Directory in an untrusted domain.
    b. telecommunications link.
    c. router.
    d. dial-up access line.

21. Your organization has domains in Canada, France, and Norway, wants to unite these domains in one container, but also wants to set up each domain to use the language native in its country. What Active Directory container would be appropriate in this situation?
    a. a site
    b. a forest
    c. a parent domain
    d. a tree
22. By default an object is owned by
    a. the account that created it.
    b. the server administrator.
    c. the domain in which it was created.
    d. the OU in which it was created.
23. You have just installed Windows 2000 Server and created accounts. By default, the security for access to these accounts is via
    a. LAN Manager.
    b. Secure Socket Layer/Transport Layer Security (SSL/TLS).
    c. Kerberos.
    d. all of the above
    e. only a and c
    f. only a and b
24. What Active Directory model links two forests?
    a. joined forests
    b. separate forests
    c. single forests
    d. forests cannot be linked in any way
25. Network authentication in Windows 2000 Server as a security technique involves
    a. authorization via an account.
    b. authorization via a smart card.
    c. authorization via a one-way trust.
    d. all of the above
    e. only a and b
    f. only a and c

HANDS-ON PROJECTS

Project 4-1

In this hands-on activity, you practice installing the Active Directory to convert a standalone computer to a domain controller.

To install the Active Directory:

1. Log on to Windows 2000 Server as Administrator.
2. Click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Configure Your Server**.
3. Click **Active Directory** in the left window pane.
4. Scroll down in the right window pane and then click **Start the Active Directory Wizard**.
5. Click **Next** in the Active Directory Installation Wizard.
6. Click **Domain controller for a new domain** (notice that if this were an additional domain controller in an existing domain, you could also set it up through the Active Directory Installation Wizard by using the other option in this dialog box). Click **Next**.
7. Click **Create a new domain tree**. Click **Next**.
8. Click **Create a new forest of domain trees**. Click **Next**.
9. Enter a practice domain name, such as **mycompany.com**, or enter a name provided by your instructor. Click **Next** (at this point the wizard will take a few moments to check that a domain does not already exist and to set up the domain).
10. Leave MYCOMPANY (or the name you entered in Step 9, if it is different) as the domain NetBIOS name (for users of other versions of Windows) and click **Next**.
11. Leave C:\WINNT\NTDS as the database location and C:\WINNT\NTDS as the log location (your system may use a different drive location by default, depending on the location of the \WINNT folder). Click **Next**.
12. Leave the shared system volume location as the default, C:\WINNT\SYSVOL (again the drive will depend on the location of the \WINNT folder). Notice that the wizard warns that the Sysvol folder must be on a volume formatted with NTFS version 5. Click **Next**. (At this point, if a DNS server is not already installed or available on the network, the wizard provides an option to install it as part of the setup process.)
13. The Permissions dialog box gives you the option to use permissions compatible with pre-Windows 2000 servers, such as Windows NT Server 4.0—or to use permissions compatible only with Windows 2000 servers. Click **Permissions compatible with pre-Windows 2000 servers** for this project. (Note that this is also a good choice if there are Web servers in the domain, because they may interact with non-Windows 2000 servers.) Click **Next**.
14. Enter an Administrator password and confirm it for use in the Directory Services Restore Mode. Click **Next**.

15. Review in the summary scroll box the selections you have made and note them in your lab journal or in a word-processed document. Is there an option to retrace your steps in case you want to change a parameter? Click **Next** to proceed. (If you do not have permission from your instructor to install the Active Directory, click Cancel at this point to exit setup.)
16. Wait a few minutes as the wizard configures the Active Directory. Notice the line near the bottom of the dialog box that shows each configuration activity.
17. Click **Finish**.
18. Make sure you have saved any open work, and then click **Restart Now**.

Project 4-2

In this assignment, you view the tool used to manage domains and trust relationships.

To view the domain management tool:

1. Click **Start**, point to **Programs**, and point to **Administrative Tools**.
2. Click **Active Directory Domains and Trusts**.
3. In the left window pane, right-click the domain you created in the last project (or an existing domain).
4. Click **Properties**.
5. Click each of the **General**, **Trusts**, and **Managed By** tabs to view their contents. Make notes about their contents in your lab journal or in a word-processed document.
6. Click **Cancel**.
7. Close the Active Directory Domains and Trusts manager.

Project 4-3

In this assignment, you view the tool used to manage Active Directory users and computers.

To view the user and computer management tool:

1. Click **Start**, point to **Programs**, and point to **Administrative Tools**.
2. Click **Active Directory Users and Computers**.
3. In the left window pane, right-click the domain you created (or an existing domain).
4. Click **Properties**.
5. Click each of the **General**, **Managed By**, and **Group Policy** tabs to view their contents. Make notes about their contents in your lab journal or in a word-processed document.
6. On the Group Policy tab, click the **Default Domain Policy**, and then click **Properties**.
7. Click the **Security** tab.
8. Click each of the groups listed in the Name box and watch to see what permissions are granted to each (record your observations).

9. How can you add another group to the list?
10. Click **Cancel** in the Default Domain Policies Properties dialog box.
11. Click **Cancel** in the domain properties dialog box.
12. Close the Active Directory Users and Computers management tool for the domain.

Project 4-4

In this hands-on activity you install the Security Templates snap-in and find out where to view the security set up in the default template for services.

To install the Security Template:

1. Click **Start** and **Run**.
2. Click the **Browse** button to find the MMC at \WINNT\system32\mmc.exe.
3. Once you find mmc.exe, double-click it and click **OK** to start it in the Run dialog box. Maximize the MMC windows, if necessary.
4. Click the **Console** menu and then click **Add/Remove Snap-in**.
5. Click **Add** in the Add/Remove Snap-in dialog box.
6. Use the scroll bar to find **Security Templates**, click this option, and then click **Add**.
7. Click **Close**, and then click **OK**.

To view the default security for services:

1. Double-click **Security Templates** in the left pane of the MMC and then double-click C:\WINNT\Security\Templates (your drive and directory location may be different).
2. Double-click the **setup security** template (in the left or right pane) and then click **System Services** under it.
3. Scroll through the services listed and make a note of 5 or 10 services.
4. Double-click the **Computer Browser** service and notice which startup mode it uses.
5. Click the **Edit Security** button.
6. Notice which groups have access to this service.
7. What permissions are given to these groups?
8. Record your observations in your lab journal or in a word-processed document.
9. Click **Cancel** and click **Cancel** again.
10. Leave the MMC open for the next project.

Project 4-5

In this project, you view the default account policies in Windows 2000 Server.

To view the default account policies:

1. Go back to the MMC that you left open in Hands-on Project 4-4.
2. Find the setup security template in the left pane and click **Account Policies** under it.

3. Double-click **Password Policy** in the right window pane.
4. What attributes are available for the password policy and how are they set? Record this information in your lab journal or in a word-processed document.
5. Double-click **Maximum Password Age**.
6. Notice that you can change the expiration period for passwords in the template (however this only changes the template, which would have to be imported again to a group policy).
7. Click **Cancel**.
8. Close the MMC (if you are asked whether to save the console settings, click No).

CASE PROJECT

Aspen Consulting Project: Planning the Active Directory and Security

Moose Jaw Outfitters, the company with which you worked in the last chapter, is contacting you again to help them plan the Active Directory and security for their networks in Winnipeg, Canada and St. Cloud, Minnesota. The Winnipeg site has a medium sized-network of 170 users, and 5 subnets using TCP/IP. That site also has Internet connectivity and maintains a Web server. Both sites have Customer Service, Business, Inventory, and IT Department members. The Winnipeg site houses most of the management team, but St. Cloud has a vice president in charge of that location, an operations manager, and supervisors for the Customer Service, Business, Inventory, and IT groups at that location. Winnipeg also has a Marketing Department.

1. The IT Department is very inexperienced with the Active Directory. Create a brief presentation for them about the Active Directory, including its purpose, Active Directory elements, how it is backed up, and the information contained in the Active Directory.
2. Prepare a list of questions for the company that would help you in planning their Active Directory implementation.
3. Before you have a chance to complete your research, the company asks for your preliminary evaluation of how the Active Directory might be implemented. On the basis of what you already know about this company, explain how you would use the following Active Directory elements and why:
   - OUs
   - Domains
   - Trees
   - Forests
   - Sites

4. Where would you place DCs and global catalog servers in this implementation?
5. Create a presentation of the Windows 2000 Security options that will be useful for Moose Jaw Outfitters.
6. Explain for the IT Department how IPSec might be an advantage for the company as a way to secure communications in sensitive areas, such as for the Business and Marketing Departments.
7. What type of security do you recommend for the Web server? Why?

Optional Case Projects for Teams

Team Case One

The consultants at your company are interested in how the Windows 2000 Active Directory compares to directory services in Windows NT 4.0 Server. Create a team to research the similarities and differences and present your findings.

Team Case Two

The consultants are also interested in the emerging Kerberos developments and how they can be of benefit to Windows 2000 Server users. Form a team to research Kerberos to explain how it works, its benefits, and which new Kerberos implementations are emerging.